# Atlas

## Cross-Platform C++ Bitcoin Wallet

### Philip Glazman

#### Spring 2018

# Table of Contents

1. Introduction
2. Installation Instructions
3. User Manual
4. Design
5. Insights and Challenges
6. Conclusion
7. Bibliography
8. Source Code
9. Test Cases

# Introduction

##### Glossary

* Bitcoin – Network protocol used to reach consensus on who owns bitcoins.
* bitcoin – The value transferred in the Bitcoin protocol.
* Satoshis – The lowest denominator of bitcoin. One satoshi is 1/100millionth of a bitcoin.
* BIP – Bitcoin Improvement Proposals (BIP) are approved or pending proposals to the Bitcoin protocol. Several BIPs provide a standard for how the protocol or nodes should behave. This project uses several BIP standards regarding how wallets ought to be implemented.
* Mining – The process by which the network reaches consensus and a transaction is confirmed.
* Script – The programming language used by Bitcoin to write scripts. This language operates uses operation codes on a reverse polish notation stack.
* Smart Contract – A piece of code that is self-enforcing on the blockchain.
* Blockchain – Public data structure that maintains a ledger representing the entire state of the network.

For more information on these terms and others not covered, please consider reviewing the open-source Bitcoin wiki located here: <https://en.bitcoin.it/>

##### Background

Bitcoin is a digital money developed in 2009 where each node participating in the network can independently validate transactions and propagate them throughout the network using software similar to bittorrent. The protocol relies on public-key cryptography to create public addresses for the end-user. In terms of bitcoin, a wallet software manages the private keys that are associated with each public address. These keys gives users ownership in spending transactions and bitcoin. In bitcoin, the wallet is an abstraction that allows the end-user to send and receive payments.

##### Project

The aim of this project is to create a user-friendly bitcoin wallet implementation that encourages self-ownership of bitcoins and the use of bitcoin’s Script language. Several Bitcoin wallets exist in the ecosystem but there does not exist a wallet that provides an abstraction layer that allows the end-user to interact with bitcoin smart contracts in a user-friendly way.

Atlas proposes a different way for the user to interact with how interact with Bitcoin. With a focus on financial independence through education, Atlas provides a straightforward way for the user to write smart contracts and learn more about the underlying low-level protocol.

##### Warning

This wallet was not extensively tested for security vulnerabilities, therefore should not be used with real bitcoin. The current implementation of Atlas operates on the Bitcoin test network and uses test network bitcoins. Funds sent and received should be used with addresses that have a testnet prefix.

##### Other Notes

This project heavily relied on Andreas M. Antonopoulos’s *Mastering Bitcoin*, open-source documentation notes on Libbitcoin on the Libbitcoin Wiki, and Aaron Jaramillo’s tutorials on Libbitcoin. These, among other scattered documentation along the web, were very helpful and resourceful. Several illustrations are used and referenced in this documentation that are from *Mastering Bitcoin*.

# Installation Instructions

Libbitcoin

Curl

Boost

JsonCPP

Qt

All with cmake

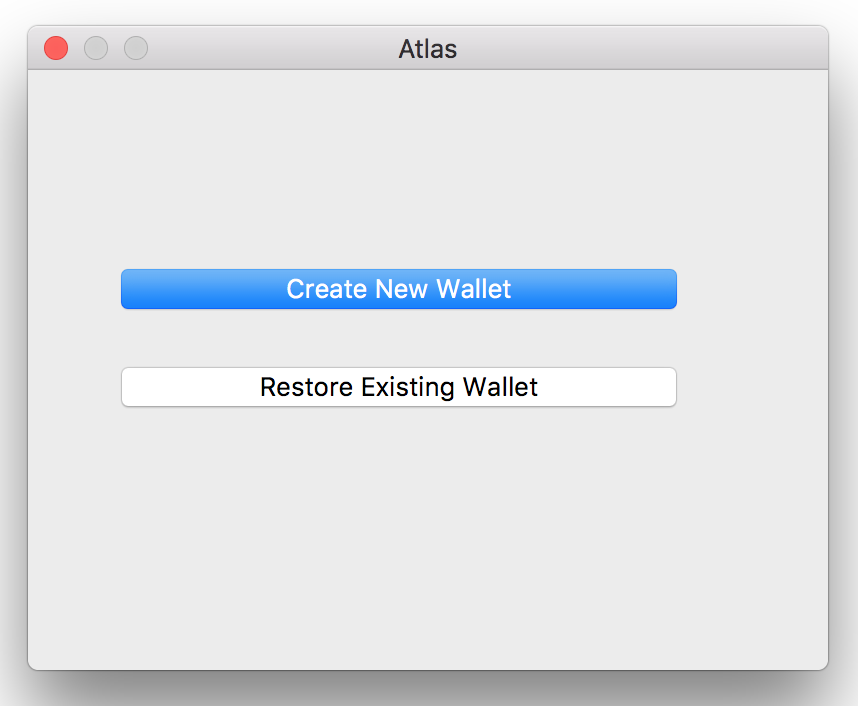
# User Manual

##### Warning

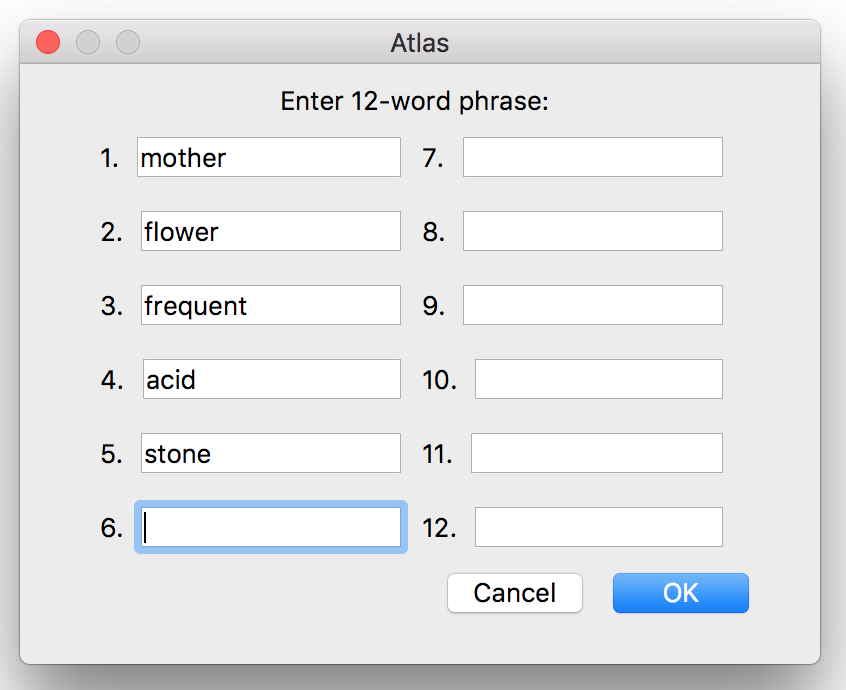
Atlas uses industry standards for wallet management but there are several risks involved. As noted in the warning section in the Introduction, this wallet should not be used with real bitcoin.

##### Wallet Creation

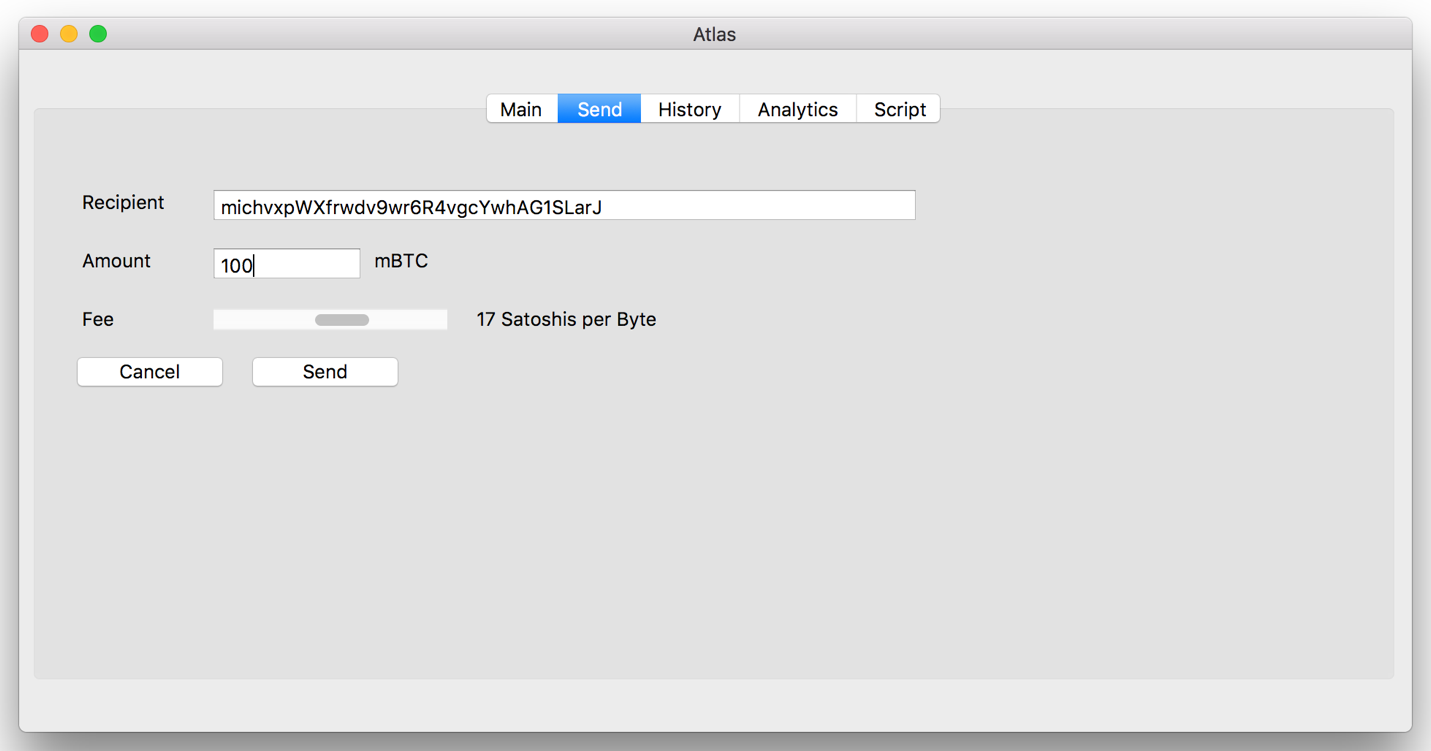
Upon starting the wallet, the user is prompted with a choice to either start a new wallet or restore an existing wallet. When a user starts a new wallet, a new seed is created that maintains the wallet. When a user chooses to restore a wallet, Atlas prompts the user for twelve words that comprises the mnemonic phrase. The user should keep these mnemonic words secret as they are the key to the wallet.



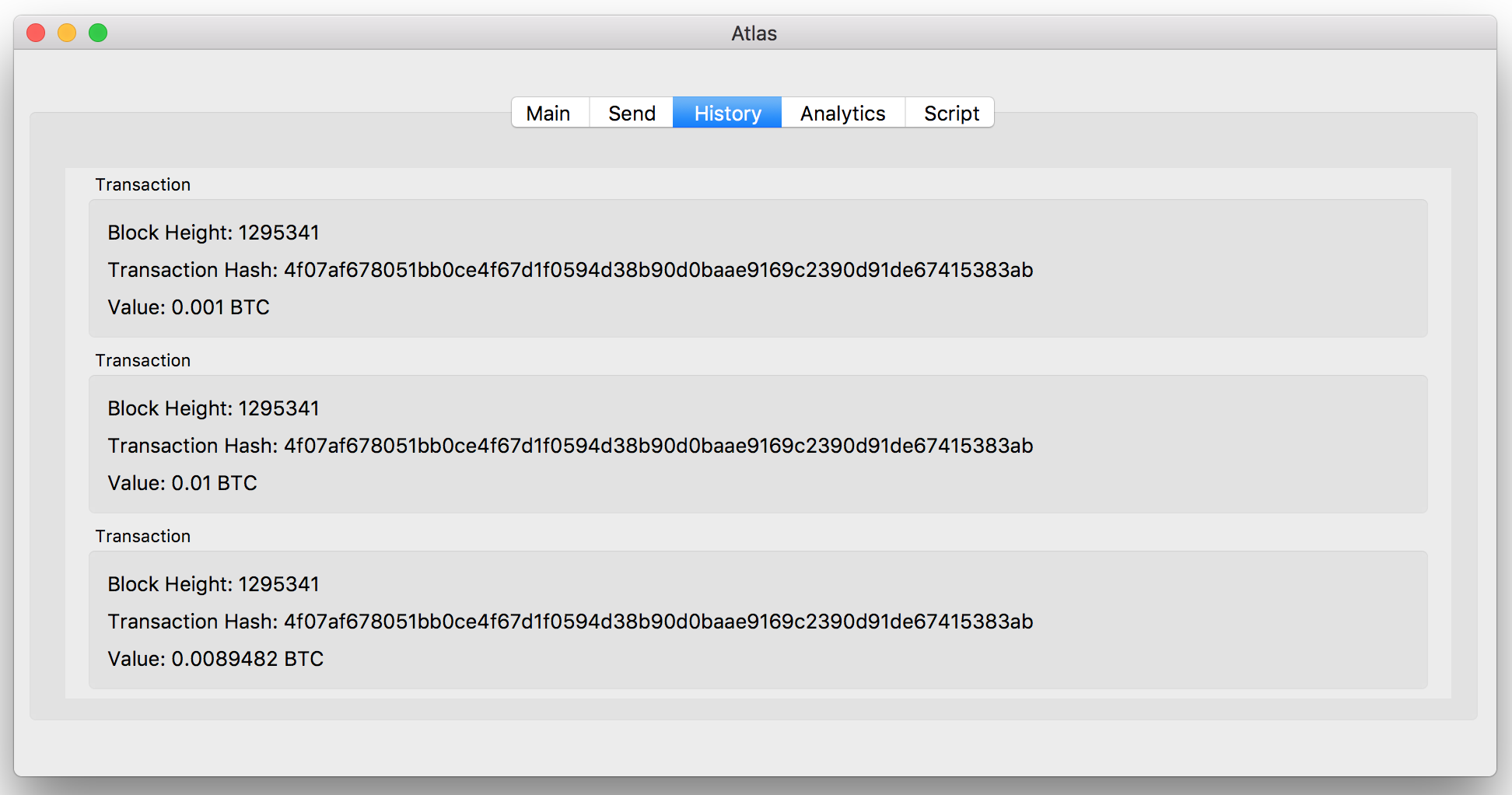
##### Mnemonic Phrase



##### Sending Bitcoin



##### Transaction History

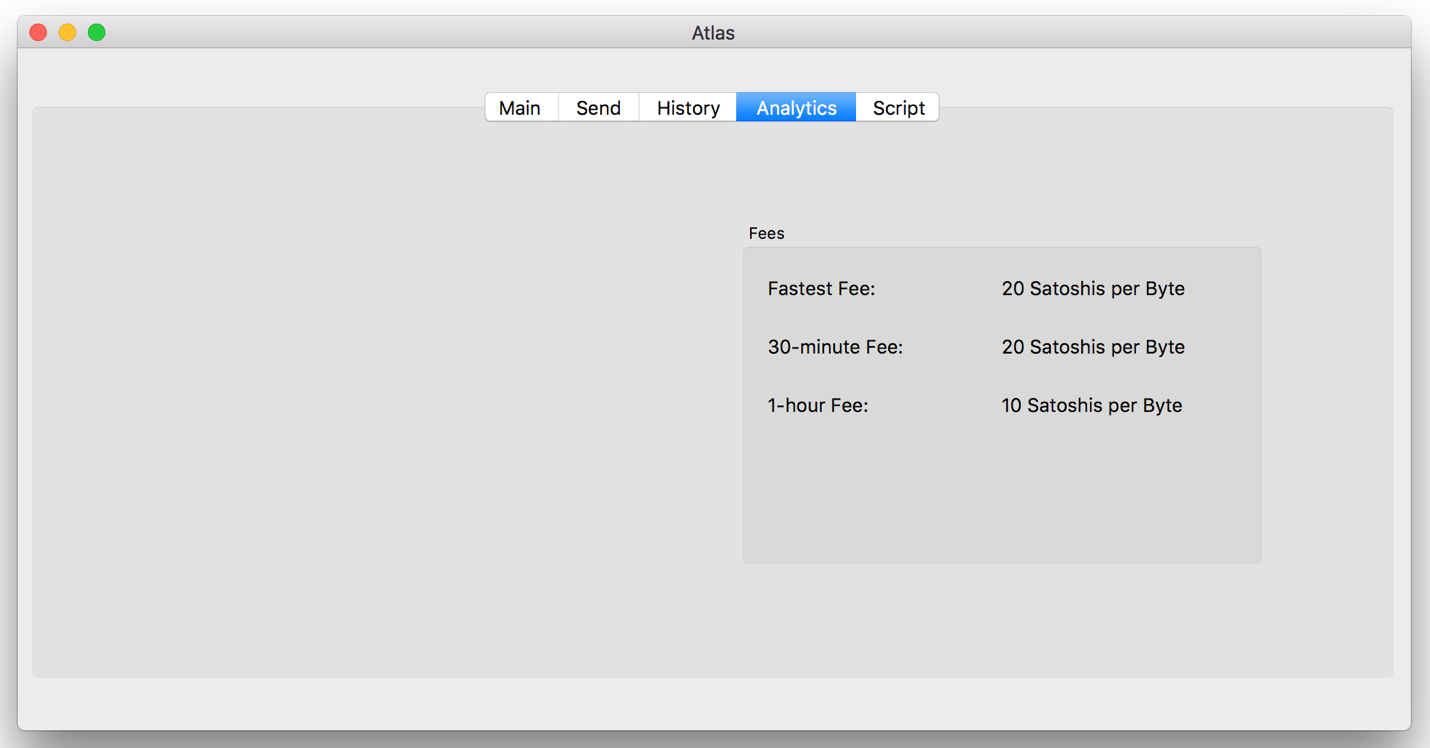


##### Understanding Network Fees

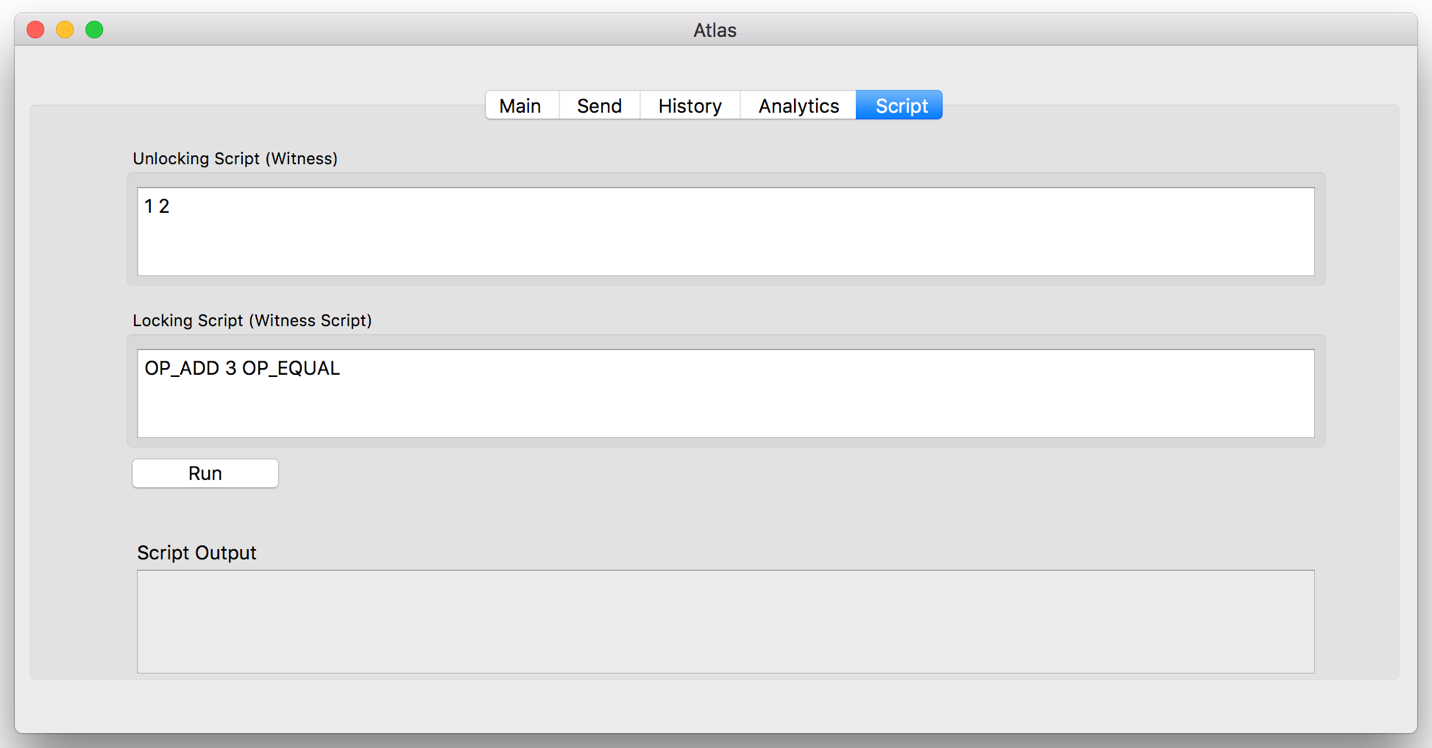
Transactions sent through the Bitcoin network may include a fee. The fee is optional and is set by the sender. A higher fee signals to the network that a specific transaction should be given priority for confirmation while a lower fee can lead to longer confirmation times. The fee market in Bitcoin is a free market set by supply and demand. As a result, the fee market changes over time and transactions can cost differently over the course of a day.

Atlas in its current implementation uses bitcoinfees.earn.com, a third-party API, to receive a suggested transaction fee. Using this API, Atlas is able to suggest to the user three fee costs located in the Analytics tab of Atlas. The user can choose to ignore these suggestions and selected a different fee.

Fees are represented as Satoshi per Byte. This measurement informs the user that it costs n Satoshi per byte. If a transaction is 250 bytes in size and a suggested fee is 40 Satoshis per Byte, the fee will be 250\*40 Satoshis.



##### Using Bitcoin Script



# Design